

REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-23 are pending in the application. The Examiner additionally stated that claims 1-23 are rejected. By this amendment, claims 1, 3-12, 14-20, and 22-23 have been amended. Hence, claims 1-23 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Claims

Claim Objections

The Examiner objected to claim 22 because of it states “plurality coprocessors” when it should read “plurality of coprocessors.” Applicant notes that claim 22 has been amended to read accordingly and therefore requests that the Examiner withdraw his objection to the claim.

Rejections Under 35 U.S.C. §103

The Examiner rejected claims 1-8 and 22 under 35 U.S.C. 102(a) as being unpatentable over Moyer (5,983,338) in view of Strongin (6,559,850 B1). Applicant respectfully traverses the Examiner’s rejections.

With regard to claim 1, the Examiner noted that Moyer has disclosed an interface (figure 1, element 30) for transferring data between a central processing unit (CPU) (figure 1, element 12) and a plurality of coprocessors (figure 1, elements 14 and 16), the interface comprising:

- i. an instruction bus (column 6, line 34 and figures 2 and 3, element 61), configured to transfer instructions to a plurality of coprocessors in an instruction transfer order (an order is inherent), wherein particular instructions direct designated ones of the plurality of coprocessors to transfer data to/from the CPU (figures 22-26, UU field); and

- ii. a data bus (column 8, lines 65-66 and figures 2 and 3, element 72), coupled to said instruction bus (since both go from processor to coprocessor, they are coupled), configured to subsequently transfer the data.

The Examiner furthermore stated that Moyer does not disclose wherein data order signals within said data bus prescribe a data transfer order that differs from said instruction transfer order, but that Strongin has disclosed in figures 3 and 4 a read retrieval order that differs from the read request order. The Examiner noted that an instruction as in Moyer is essentially a request for data or a read request. When the data is sent, that is a read retrieval. The Examiner remarked that the figure shows signals (identifier) that indicate the order of the data, and that Strongin has shown in column 6, lines 36-44 that this difference in ordering allows for data accesses to be quicker, and thus, the quickness of data access would have motivated one of ordinary skill in the art to modify the design of Moyer to include the out of order data retrieval disclosed by Strongin. The Examiner further opined that with this modification in place, the data order signals would prescribe transfer of a data element corresponding to a specific outstanding instruction relative to all outstanding instructions and since the disclosure of Moyer is dealing with transfer of data between a processor and coprocessors, the data is inherently associated with outstanding instructions and further it is inherent that the instruction is relative to all other instructions in some manner. The Examiner noted that this relationship could simply be instruction order, which would mean that the outstanding instruction is relative to the other instructions in that some are before it and some are after in program order. The Examiner concluded that it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the design of Moyer to retrieve data out of order as taught by Strongin so that data accesses can be achieved quicker.

Claim 1 as amended is provided below for ease of reference.

1. An interface for transferring data between a central processing unit (CPU) and a plurality of coprocessors, the interface comprising:
 - an instruction bus, configured to transfer instructions to the plurality of coprocessors in an instruction transfer order, wherein particular instructions direct one of the plurality of coprocessors to transfer the data to/from the CPU; and
 - a data bus, coupled to said instruction bus, configured to transfer the data, wherein data order signals within said data bus specify a data transfer order that differs from said instruction transfer order, and wherein said data order signals specify transfer of a data element corresponding to a specific outstanding instruction that is relative in order to all outstanding instructions, said outstanding instructions being those of said particular instructions transferred to said one of the plurality of coprocessors that have not completed a data transfer;

wherein the interface keeps track of an order of said outstanding instructions, and wherein said data order signals indicate said order, and wherein said data order signals are provided with said data element as said data element is transferred.

In combination, claim 1 recites an instruction bus and a data bus that provide an interface for transferring data between a CPU and a plurality of coprocessors. The instruction bus transfers instructions to the plurality of coprocessors in an instruction transfer order, where particular instructions direct one of the plurality of coprocessors to transfer the data to/from the CPU. The data bus transfers the data in a data transfer order that differs from the instruction transfer order. Data order signals within the data bus specify transfer of a data element corresponding to a specific outstanding instruction that is relative in order to all outstanding instructions. In addition, the claim recites that the *outstanding* instructions are those of the particular instructions transferred to the one of the plurality of coprocessors that have not completed a data transfer. Furthermore, the claim recites

that the interface keeps track of an order of the outstanding instructions, and that the data order signals indicate the order, and that the data order signals are provided with the data element as the data element is transferred.

Applicant has studied the teachings of both Moyer and Strongin and finds that both teach the transfer of data to/from a coprocessor (or, in the case of Strongin, the transfer of read requests to an AGP-enabled Northbridge). Moyer teaches in-order data transfer. Strongin teaches out-of-order data transfer where a transaction ID is associated with each individual data transaction. Figures 3-8 and associated descriptions in Strongin teach that each transaction ID is associated with individual data transactions within a number of AGP pipelined transactions, to identify the individual transactions. The association of a transaction ID with the individual transactions includes associating a transaction ID with each individual read request within a number of AGP pipelined read requests and associating *an identical* transaction ID with each individual data unit, within a number of pipelined data units, corresponding to each individual memory request within the number of AGP pipelined memory requests. (Abstract) In fact, referring to Figure 4 and discussing how data is delivered out of order, Strongin states that pending requests are serviced out of order, and that since read request N 100N is to row 4 of DRAM 130, transaction id N 300N and its associated data 200N are shown following read request 1 1001 and its associated data 2001, which means that read request N 100N *was serviced* immediately subsequent to read request 1. (col. 10, lines 46-58). To correlate the teachings of Strongin to those of Applicant, those read requests of Strongin *which have not been serviced* are *outstanding*. Once a read request has been serviced (i.e., its data has been transferred), then it is no longer *outstanding*. Furthermore, it is clear from the teachings of Strongin that his transaction IDs are generated at the time of a read request as opposed to generation at the time of data transfer, for he writes, “[i]n such a system, a transaction (transaction id) is associated both with each read request and with the data unit returned in response to such read request. (Col. 10, lines 26-29) Moreover, Strongin’s transaction IDs do not indicate a transfer that is relative in order to outstanding instructions. Rather, Strongin’s transaction IDs indicate an absolute order upon transfer of the data that indicates the ordering of transactions per previously issued requests.

Thus, he teaches assigning a transaction ID that indicates the order of a read request when initiated, for he writes: “[d]epicted is that each read request 1001-100N now has associated 300 with its transaction ID 3001-300N indicative of the relative order of read requests 1001-100N.” (Col. 10, lines 32-34).

Applicant’s invention, on the other hand, provides for data order signals within a data bus that specify a data transfer order that differs from an instruction transfer order, and moreover that the data order signals specify transfer of a data element corresponding to a specific outstanding instruction *that is relative in order to all outstanding instructions*, where it is expressly claimed that *outstanding instructions* are those of the particular instructions that have not completed a data transfer. Thus, Applicant’s data order signals do not indicate the order of transactions when initiated, but rather indicate the order of transfer of a data element corresponding to a specific outstanding instruction that is relative in order to those instructions that have not completed a data transfer. For example, refer to Table 2 on page 36 of the instant application. The value of data order signals CP_torder_m are assigned, as shown in the table, relative in order to the oldest outstanding data transfer, and are in no way indicative of the order in which the transactions were initiated. Data order signals CP_forder_m, also presented in Table 2, are assigned similarly. Furthermore, Applicant’s invention provides an interface that keeps track of the order of the outstanding instructions, and that the data order signals indicate the order, and that the data order signals are provided with the data element as the data element is transferred. Applicant has searched the teachings of both Moyer and Strongin and finds that neither of the inventors, separately or in combination, provide any motivation whatsoever to one skilled in the art to provide apparatus or method having data order signals that specify transfer of a data element corresponding to a specific outstanding instruction *that is relative in order to all outstanding instructions*, where the *outstanding instructions* are those particular instructions transferred to the one of the plurality of coprocessors that have not completed a data transfer, and which moreover provides an interface that keeps track of the order of the outstanding instructions, where the data order signals indicate the order of the outstanding instruction, and where the data order signals are provided with the data element as the data element is transferred. In

fact, the transaction ids of Strongin are 1) generated at the time of a read request; and 2) are indicative of the order of issued instructions, but 3) are not indicative of the order of outstanding instructions. For example, referring to Figure 4 of Strongin, after servicing (i.e., completing transfer of associated data) read requests 1 and N, which come in out of order, the transaction id for read request 2 does not prescribe transfer of a data element corresponding to a specific outstanding instruction that is relative in order to all outstanding instructions (i.e., read request 2, read request 3, and read request 4); it only prescribe transfer of a data element corresponding to a specific outstanding read request that is relative to all dispatched read requests—both those with have been serviced and those which are outstanding. Hence, a significant disadvantage of Strongin's enumeration scheme is that, in this example, the memory controller 200 cannot reuse, say, transaction id 4 in a read request until read request 4 is serviced. In contrast, as shown in one embodiment of the present invention described in Table 2 on page 36 of the instant application, Applicant's data order signals (i.e., CP_torder_m) specify which *outstanding* instruction data transfer is for in terms of "oldest," "2nd oldest," etc., thus allowing reuse of all specific instruction order designators that are not outstanding. Accordingly, Strongin's approach requires extensive intelligence in the memory controller to ensure that data is reordered.

Applicant moreover notes that Strongin teaches away from prescribing transfer of a data element corresponding to a specific outstanding instruction that is relative in order to all outstanding instructions by noting "since the transaction ids 3001-300N are to be denoted by the use of 3 data lines, or 3 bits, the greatest number of transaction ids that can be established is 8." (col. 12, lines 5-8) In summary, Strongin teaches assignment of a transaction ID to a data element that is identically fixed to a corresponding AGP read request and which is thus provided at the time of the read request, not at the time of data transfer. In contrast, Applicant's invention claims data order signals that specify transfer of a data element corresponding to a specific outstanding instruction *that is relative in order to all outstanding instructions*, where the *outstanding* instructions are those of the particular instructions transferred to said one of the plurality of coprocessors that have not completed a data transfer, and moreover where the interface keeps track of the order

of the outstanding instructions, and where the data order signals indicate the order of the outstanding instructions, and where the data order signals are provided with the data element as the data element is transferred.

In summary then, neither, Moyer nor Strongin provide any teaching or motivation that would lead one skilled in the art to provide for data order signals that specify transfer of a data element corresponding to a specific outstanding instruction *that is relative in order to all outstanding instructions*, where the *outstanding* instructions are those of the particular instructions transferred to the one of the plurality of coprocessors that have not completed a data transfer. Nor do either of the two references provide teachings that even hint at an interface that keeps track of the order of the outstanding instructions, data order signals indicative of the order of the outstanding instructions, or provision of the data order signals along with the data element as the data element is transferred. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejection of claim 1.

With respect to claims 2-8, these claims depend from claim 1 and add further limitations that are neither anticipated nor made obvious by Moyer, Strongin, or Moyer and Strongin in combination. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 2-8.

Claim 22 is provided below for ease of reference.

22. A method for transferring data between a CPU and a plurality of coprocessors, the method comprising:

transmitting instructions to one of the plurality of coprocessors, each of the instructions directing a data transfer between the CPU and the one of the plurality of coprocessors, wherein said transmitting is provided in a specific instruction order; and

transferring the data in an order different from the specific instruction order, said transferring comprising:

specifying transfer of a data element corresponding to a specific outstanding instruction that is relative in order to all outstanding instructions, the outstanding instructions being those of the instructions transmitted to the one of the plurality of coprocessors that have not completed a data transfer; and

providing the data order signals with the data element as the data element is transferred.

In arguments drawn from the same points as made with reference to the rejection of claim 1, the Examiner provided his reasons for rejecting claim 22. In response, Applicant asserts that claim 22 recites, in combination with other method elements “keeping track of the order of outstanding instructions and specifying transfer of a data element corresponding to a specific outstanding instruction that relative in order to all of the outstanding instructions, the outstanding instructions being those of the instructions that have not completed a data transfer.” The “outstanding instructions being those of the instructions transmitted to the one of the plurality of coprocessors that have not completed a subsequent data transfer” limitation, which is similar to the limitation discussed above in traversal of the rejection of claim 1, is neither taught nor suggested by Moyer, Strongin, or Moyer and Strongin in combination. Neither is the limitation “keeping track of the order of outstanding instructions” taught or even suggested by Moyer or Strongin. This is because Strongin’s transaction ids are assigned at the time of a read request. Consequently, there is no need for Strongin to keep track of the relative order. Applicant therefore respectfully asks that the rejection of claim 22 be withdrawn.

The Examiner rejected claims 9 and 23 under 35 USC 103(a) as being unpatentable over Moyer in view of Strongin as applied to claims 1-8 and 22, and further in view of Hennessy. Applicant respectfully traverses. And in that Applicant has argued over the combination of Moyer’s and Strongin’s teachings with reference to the rejections of claims 1 and 22, and noting that claim 9 depends from claim 1 and adds further limitations, it is respectfully requested that the Examiner withdraw his rejection of claim

9. Likewise, since claim 23 depends from claim 22 and adds further limitations, Applicant respectfully requests the withdrawal of the rejection of claim 23.

The Examiner rejected claims 10-12 and 14-20 under 35 USC 103(a) as being unpatentable over Moyer in view of Tanenbaum and further in view of Strongin. Applicant respectfully traverses the Examiner's rejections.

With regard to claim 10, the Examiner stated that Moyer discloses a computer program product for use with a computing device, the computer program product comprising:

1. a computer usable medium, for causing a coprocessor interface to be described that transfers data between a CPU and a plurality of coprocessors, said computer readable program code comprising:
 - (1) an instruction bus, said instruction bus configured to transfer instructions to said plurality of coprocessors in an instruction transfer order, wherein particular instructions direct designated ones of the plurality of coprocessors to transfer said data to/from said CPU; and
 - (2) a data bus, said data bus configured to subsequently transfer said data.

The Examiner remarked that Moyer does not disclose having computer readable program code embodied in said medium and first program code and second program code. The Examiner added that Moyer does not disclose wherein said data order signals within said data bus prescribe a data transfer order that differs from said instruction transfer order. The Examiner stated that Tanenbaum has disclosed on pages 10-12 that hardware is logically equivalent to software and that the boundaries between them are fluid, noting that Strongin has disclosed in figures 3 and 4 a read retrieval order that differs from the read request order. The Examiner stated that an instruction in Moyer is essentially a request for data or a read request. When the data is sent, that is a read retrieval. The Examiner remarked that the figure shows signals (identifier) that indicate the order of the data.

In addition, the Examiner noted that Tanenbaum has shown on page 11 that for one factor involved in deciding whether to implement a function in hardware or software is frequency of change, stating that it is easier to change software code than to change the layout of a hardware system. The Examiner remarked that this ease of change would have motivated one of ordinary skill in the art to modify the design of Moyer to implement the disclosed apparatus as program code taught by Tanenbaum. It was noted that Strongin has shown in column 6, lines 36-44 that this difference in ordering allows for data accesses to be quicker, and that this quickness of data access would have motivated one of ordinary skill in the art to modify the design of Moyer to include the out of order data retrieval disclosed by Strongin. In addition, the Examiner noted that with this modification in place, the data order signals would prescribe transfer of a data element corresponding to a specific outstanding instruction relative to all outstanding instructions and since the disclosure of Moyer is dealing with the transfer of data between a processor and coprocessors, the data is inherently associated with outstanding instructions and further it is inherent that the instruction is relative to all other instructions in some manner. The Examiner pointed out that this relation could simply be instruction order, which would mean that the outstanding instruction is relative to the other instructions in that some are before it and some are after in program order.

The Examiner concluded that it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the design of Moyer to implement his design in program code as taught by Tanenbaum and to retrieve data out of order as taught by Strongin so that data accesses can be achieved quicker and so that changes may be made easier.

Claim 10 is provided below for ease of reference.

10. A computer program product for use with a computing device, the computer program product comprising:
 - a computer usable medium, having computer readable program code embodied in said medium, for causing a coprocessor interface to be described that transfers data between CPU and a plurality of coprocessors, said computer readable program code comprising:
 - first program code, for providing an instruction bus, said instruction bus configured to transfer instructions to said plurality of coprocessors in an instruction transfer order, wherein particular instructions direct one of said plurality of coprocessors to transfer said data to/from said CPU; and
 - second program code, for providing a data bus, said data bus configured to transfer said data, wherein data order signals within said data bus specify a data transfer order that is different from said instruction transfer order, and wherein said data order signals specify transfer of a data element corresponding to a specific outstanding instruction that is relative in order to all outstanding instructions, said outstanding instructions being those of said particular instructions transferred to said one of said plurality of coprocessors that have not completed a data transfer;
 - wherein the coprocessor interface keeps track of an order of said outstanding instructions, and wherein said data order signals indicate said order, and wherein said data order signals are provided with said data element as said data element is transferred.

In transversal of the Examiner's rejection of claim 10, Applicant responds that claim recites, in combination with other elements "second program code, . . . , wherein said data order signals specify transfer of a data element corresponding to a specific outstanding instruction that is relative in order to all outstanding instructions, said

outstanding instructions being those of said particular instructions transferred to said one of said plurality of coprocessors that have not completed a data transfer.” This limitation, similar to that recited in claim 1, and similar to the method element recited in claim 22, is neither taught nor suggested by Moyer, Strongin, or Moyer and Strongin in combination. Likewise, the limitation “wherein the coprocessor interface keeps track of an order of said outstanding instructions, and wherein said data order signals indicate said order, and wherein said data order signals are provided with said data element as said data element is transferred,” is not present nor are there any teachings in Strongin, Moyer, or Moyer and Strongin in combination that would motivate one skilled in the art to provide such novel features. Furthermore, although Tanenbaum may suggest that hardware is logically equivalent to software and that the boundaries between hardware and software are fluid, a motivation to implement, in program code rather than hardware, data order signals that prescribe transfer of a data element corresponding to a specific outstanding instruction that is relative in order to all outstanding instructions, is absent apart from a motivation or other suggestion to do so in hardware. Accordingly, Applicant respectfully requests that the rejection of claim 10 be withdrawn.

With regard to claims 11-12, these claims depend from claim 10 and add further limitations that are neither anticipated nor made obvious by Moyer, Strongin, Tanenbaum, or any combination of the three aforementioned references. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections of claims 11-12.

Claim 14 is provided below for ease of reference.

14. A computer data signal embodied in a transmission medium, the computer data signal comprising:

computer-readable first program code, for providing an instruction bus for transferring instructions to a plurality of coprocessors in an instruction transfer order, wherein particular instructions direct a particular coprocessor to transfer data to/from a CPU; and

computer-readable second program code, for providing a data bus for transferring said data, wherein data order signals within said data bus specify a data transfer order that differs from said instruction transfer order, and wherein said data order signals specify transfer of a data element corresponding to a specific outstanding instruction that is relative in order to all outstanding instructions, said outstanding instructions being those of said particular instructions transferred to said particular coprocessor that have not completed a data transfer;

wherein said data order signals indicate an order of said outstanding instructions, and wherein said data order signals are provided with said data element as said data element is transferred.

In arguments drawn from the same points as made with reference to the rejection of claim 10, the Examiner provided his reasons for rejecting claim 14. In response, Applicant asserts that claim 14 recites, in combination with other elements, “computer-readable second program code, . . . wherein said data order signals specify transfer of a data element corresponding to a specific outstanding instruction that is relative in order to all outstanding instructions, said outstanding instructions being those of said particular instructions transferred to said particular coprocessor that have not completed a data transfer.” This limitation, similar to the limitations discussed above in traversal of the rejection of claim 10, are neither taught nor suggested by Moyer, Strongin, Tanenbaum, or any combination of the three references. In addition, Moyer, Strongin, and Tanenbaum utterly fail to teach the limitation “wherein said data order signals indicate an order of said outstanding instructions, and wherein said data order signals are provided with said data element as said data element is transferred.” Accordingly, Applicant respectfully asks that the rejection of claim 14 be withdrawn.

With regard to claims 15-20, these claims depend from claim 14 and add further limitations that are neither anticipated nor made obvious by Moyer, Strongin, Tanenbaum, or any combination of the three aforementioned references. Accordingly,

Applicant respectfully requests that the Examiner withdraw his rejections of claims 15-20.

The Examiner rejected dependent claims 13 and 21 under 35 USC 103(a) as being unpatentable over Moyer in view of Tanenbaum and further in view of Strongin as applied to claims 10-12 above, and further in view of Hennessy. Applicant respectfully traverses and directs the Examiner's attention to arguments made in traversal to the rejections of independent claims 10 and 14. Consequently, in that claim 13 depends from claim 10 and in that claim 21 depends from claim 14, and in that these claims add further limitations that are neither anticipated nor made obvious by Moyer, Strongin, Hennessy, Tanenbaum, or any combination of the four aforementioned references, Applicant respectfully asks that the rejections of claims 13 and 21 be withdrawn.

CONCLUSIONS

In view of the arguments advanced above, Applicant respectfully submits that claims 1-23 are in condition for allowance. Reconsideration of the rejections is requested, and allowance of the claims is solicited.

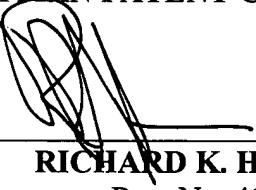
Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

EXPRESS MAIL LABEL NUMBER: **EO 004 051 039 US**
DATE OF DEPOSIT: **2/22/05**

I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 C.F.R. §1.10 on the date shown above and is addressed to Mail Stop **PETITION**, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.

Respectfully submitted,
HUFFMAN PATENT GROUP, LLC

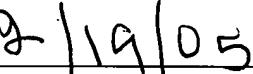
By _____


RICHARD K. HUFFMAN

Reg. No. 41,082

Tel.: (719) 575-9998

Date: _____


2/19/05